Security Onion Linux distribucija i njezine primjene

Bošnjak, Dominik

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Security Onion Linux distribucija i njezine primjene

Završni rad

Dominik Bošnjak

Osijek, 2019.

SADRŽAJ

1. UVOD

Pitanje sigurnosti možda je jedno od vječitih pitanja kako u stvarno tako i u umreženom svijetu. U današnje doba, informacije se prenose raznim sredstvima, sve više digitalnim, no koliko su sigurni ti prijenosi informacija. Uređaji koji se koriste imaju ranjivosti, slabe točke, lošu zaštitu te je samo pitanje vremena kada će informacija koja se prenosi biti ukradena, izmijenjena ili izgubljena. Pošto svi cijene vrijednost informacije, pogotovo velike korporacije, tržište je prepunjeno mnoštvom raznih sigurnosnih alata koji obećavaju, sto-postotnu, *premium* zaštitu, koje će one iste velike korporacije slijepo kupiti i postaviti, misleći da su sigurni. No kao što je navedeno ranije, samo je pitanje vremena kada će se pojaviti nova tehnika provale ili izvlačenja informacija. S druge strane je *Security Onion*, besplatna i otvorenog koda Linux distribucija koja pruža veliki broj alata i aplikacija za testiranje i uspostavljanje sigurnosti računalnih mreža. Omogućava mrežno-orijentirane (*NIDS*) i računalno-orijentirane (*HIDS*) sustave za otkrivanje upada (*IDS*) te razne alate za analizu prikupljenih podataka, no ne sprječava upade! Postavlja se pitanje kakva je uopće svrha takve distribucije? Upravo takva da navodi na pogled s druge strane, kako bi saznali što se stvarno događa s druge strane ekrana s kojeg primamo i šaljemo informacije.

Rad će biti podijeljen u tri dijela, prvi od kojih će se odnositi na teoriju, smisao mrežnog sigurnosnog nadgledanja. Drugi dio obuhvatit će proces postavljanja *Security Onion* Linux distribucije, dok će se treći dio fokusirati na mogućnosti, alate te konkretnu primjenu.

1.1 Zadatak završnog rada

Security Onion Linux distribucija sadrži veliki broj sigurnosnih aplikacija i alata, koje se mogu koristiti u svrhu zaštite i testiranja u području kibernetičke sigurnosti (primarno za sigurnosni monitoring, s obzirom da implementira sustave za otkrivanje upada). Potrebno je istražiti mogućnosti primjene Security Onion Linux distribucije, te je primijeniti za potrebe poboljšanja sigurnosti i testiranje sigurnosnih ranjivosti kroz više različitih scenarija u lokalnom mrežnom okruženju. Dobivene rezultate potrebno je analizirati, te definirati smjernice i preporuke za poboljšanje mrežne sigurnosti.

2. NAČELA RADA SECURITY ONIONA

Doug Burks započeo je Security Onion kao besplatni projekt otvorenog koda 2008. godine te je osnovao *Security Onion Solutions, LLC* 2014. godine, koji je jedini i službeni pružatelj obuke, profesionalnih usluga i hardverskih uređaja za *Security Onion.*

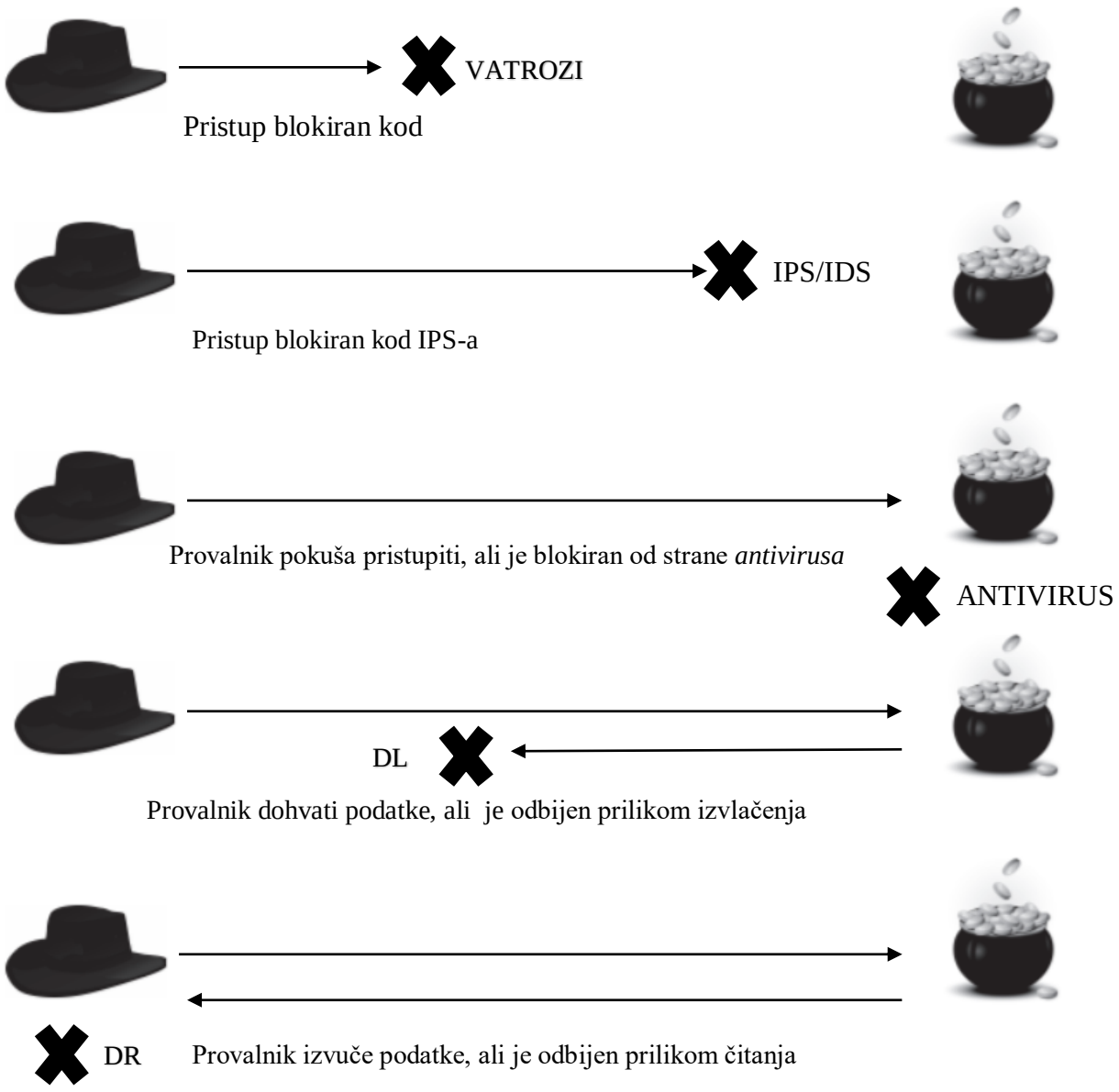
2.1 NSM

NSM – Mrežno sigurnosno nadgledanje (engl. *Network Security Monitoring*) je zaštita koja funkcionira na principu nadgledanja prometa koji prolazi kroz računalnu mrežu, skupljanja podatka o tom prometu te konačno analiza tih podataka iz koje se donose zaključci te se poduzimaju određene akcije u svrhu zaštite od neovlaštenih pristupa i radnji . Drugim riječima to je način pronalaženja provalnika u mreži i poduzimanja odgovarajućih mjera zaštite kako bi se spriječio ili smanjio štetni utjecaj neovlaštenih radnji koje se događaju ili će se dogoditi.
Dokumentacija Security Oniona [1] tvrdi kako je ključna riječ u NSM akronimu je „M“ koji stoji za nadgledanje (engl. *Monitoring)*. Podaci se mogu skupiti i analizirati ali većina malicioznih aktivnosti se ne čine maliciozna na prvi pogled. Također moguće je povući paralelu sa fizičkom sigurnosti i zamisliti NSM kao sigurnosne kamere koje postavimo na kritične točke i nakon toga nitko nije postavljen da nadgleda snimke tih kamera, što nije od prevelike koristi. Navedeno se može reći i za NSM, jer on ne sprječava neovlaštene pristupe već pruža potrebne alate kako bi ti pristupi bili pravovremeno otkriveni te adekvatno odgovoreni. Potrebno je razumjeti da su neovlašteni pristupi neizbježni, jer će odlučni provalnici s vremenom zaobići postavljenu „obranu“, no ne treba se osjećati bespomoćno. Našom voljom, strašću i željom za učenjem možemo im znatno otežati posao. Postavlja se pitanje „Ako možemo otkriti upad, zašto ga ne možemo spriječiti?“ Jednostavan odgovor bi bio da sustavi i procesi, dizajnirani radi zaštite, nisu savršeni. Mehanizmi prevencije mogu blokirati pojedine maliciozne aktivnosti, ali svakim danom pojavljuju se nove, sofisticiranije taktike koje zaobilaze te blokade.

2.1.1. NSM u usporedbi sa ostalim pristupima

Vatrozid (engl. *Firewall*), Sustavi za prevenciju/otkrivanje upada (engl. *intrusion prevention/detection system, IPS/IDS*), *antivirus* programi, DLP (engl. *data leakage protection*), DRM (engl. *digital rights managment*) sve su sustavi koji pokušavaju zaustaviti provalnike i razne

druge nametljivce. Svaki od ovih sustava je blokirajući, filtrirajući ili odbijajući (engl. *denying)* mehanizam. Njihov posao je prepoznati malicioznu aktivnost i zaustaviti je, u različitim stadijima, ovisno o mehanizmu koji se koristi. Navedeni sustavi imaju varijabilne razine uspjeha. Slika 1.1 prikazuje načine sprječavanja provalnika koji pokušava ukrasti osjetljive podatke sa nekog sustava



Slika 1.1 blokirajući, filtrirajući i odbijajući mehanizmi [2, str 10]

Na primjer česta primjedba *IDS*-a je da generira mnogo lažnih-pozitiva (engl. *false-positives)*. Također je moguća paralela sa fizičkom sigurnosti, zamislimo *IDS* kao psa čuvara. Svaki put kada se netko približi vratima njegove kuće, pas doleti i krene lajati. Bez obzira da li je na vratima poštar ili provalnik, pas reagira jednako. Isto tako i *IDS* obavještava da je netko pokušao pristupiti

promatranoj mreži, no ne daje potreban kontekst kako bi se donio zaključak da li je pristupnik bio zlonamjeran ili ne.

Za razliku od navedenih pristupa NSM je strategija koja se fokusira na vidljivost. Stvara potreban kontekst uz IDS uzbune jer uz njih kombinira potpuno hvatanje paketa (engl. *full packet capture*) i alate za analizu prikupljenih podataka.

Da bi se razumjela prednost strategije koja se fokusira na vidljivost potrebno je prvo sagledati provalu iz provalnikove perspektive. Njegov cilj nije samo provaliti, nego što duže ostati u mreži, neprimjetni i slobodni da sakupljaju informacije po želji. Ako vidljivost napravimo prioritetom u našoj mreži te ako možemo iskoristiti prednosti te vidljivost, možemo znatno otežati život provalnika.

2.2 Glavne komponente

Security Onion neprimjetno integrira i povezuje tri glavne komponente koje omogućuju NSM:

- Potpuno hvatanje paketa
- NIDS i HIDS
- Moćne alate za analiziranje

2.2.1. Potpuno hvatanje paketa

Potpuno hvatanje paketa omogućeno je putem *netsniff-ng* („švicarski nožić za mrežno upravljanje“) koji bilježi sav promet kroz mrežu i sprema ga u obliku *pcap* datoteka. *Pcap* je API za hvatanje mrežnog prometa, službeno se naziva *libpcap*, a skraćenica je potekla od izraza „packet capture“ (hvatanje paketa). Format izgleda prikazan je u tablici 1.1:

Tablica 1.1 pcap format

Globalno zaglavlje	Zaglavlje paketa	Podaci paketa	Zaglavlje Paketa	Podaci paketa	…

Prema [3]:

Globalno zaglavlje sadrži: broj verzije formata, vremensku zonu, *snaplet* – maksimalnu duljinu hvatanja, tip zaglavlja sloja podatkovnog linka.

Zaglavlje paketa sadrži: vremensku oznaku, duljinu paketa

Kada paket prođe kroz mrežu on će u potpunosti biti snimljen i spremljen, a ne samo njegovo zaglavlje s čime onda, osim što znamo tko dolazi i odlazi, znamo što sa sobom donosi ili odnosi.

2.2.2. NIDS i HIDS

Security Onion pruža višebrojne IDS opcije:

Za NIDS opcija detekcija na temelju potpisa kojom se definiraju pravila ili „potpisi/uzorci“ poznatih prijetnji i napada te se provjerava da li postoje u prometu mreže. Ovu funkcionalnost omogućuju *Snort i Suricata*. Jedan od nedostataka detekcije na temelju potpisa jest potreba za konstantnim ažuriranjem baze „uzoraka/potpisa“. Također opcija detekcije anomalija koju pruža *Bro,* on naime bilježi podatke o legitimnim korisnicima te na temelju toga traži odstupanja u odnosu na njihovo „uobičajeno ponašanje“

Za HIDS na raspolaganju su *Wazuh ili OSSEC,* koji prate i nadziru događaje na pojedinačnom uređaju u potrazi za sumnjivom aktivnošću. Omogućava analizu dnevničkih zapisa (*log analysis*), upozorenja u stvarnom vremenu, otkrivanje raznih neautoriziranih događaja.

2.2.3. Alati za analiziranje

Sa svim navedenim opcijama detektiranja i potpunim hvatanjem paketa, stvaraju se ogromne količine podataka, koje treba proučiti te razlučiti korisne od beskorisnih informacija. Na svu sreću Security Onion pruža različite i brojne alate za analiziranje koji olakšavaju taj posao, prema [1]:

Sguil – alat koji pruža jedinstveno grafičko korisničko sučelje (*GUI*), u stvarnom vremenu, koje povezuje i omogućuje istovremeni pregled više tipova upozorenja. Nadalje, kada dođe do upozorenja imamo moguće je direktno poslati *.pcap* povezan sa upozorenjem u *Wireshark* i proučiti cijelu sesiju koja je okinula upozorenje. Također omogućava komentiranje upozorenja što olakšava suradnju analitičara.

Squert – omogućava nekoliko načina vizualizacije podataka spremljenih u *Squil* bazi podataka, kao na primjer prikaz vremenskih nizova, težinski i logički grupirani skup rezultata, korištenje meta podataka. Ovo je korisno iz razloga što se drugačijim prikazom istih podatka stvara širi kontekst oko istih te je veća vjerojatnost uočavanja detalja koje obično ne bi bili primijećeni.

Kibana – omogućava brzo analiziranje i prikaz svih podataka na jednom mjestu u isto vrijeme. Vrlo je koristan alat jer pruža vizualizacije podataka raznim grafičkim prikazima koji će biti jasni čak i netreniranim osobama, što olakšava posao analitičara kada dođe do objašnjavanja rezultata klijentima.

CapME – omogućava pregled *.pcap* transkripta i skidanje .pcap datoteka. To je zapravo ono što omogućava *Kibani* i *Squertu* prikaz potpunih paketa.

2.3 Zaključak komponenti

U konačnici dostupni su *Snort* ili *Suricatu* za detekciju na temelju potpisa, *Bro* za detekciju anomalija, *Wazuh* ili *OSSEC* za računalno-orijentirani IDS, sve pokrenuto u isto vrijeme na jednom mjestu te povezano raznim alatima za analizu. Također svi alati i aplikacije koriste isto korisničko ime i lozinku koja se postavi tijekom instalacije.

3. POSTAVLJANJE I ODRŽAVANJE

Postoje dva načina postavljanja Security Oniona:

- Stand-alone – u ovome načinu sve komponente *SO-a* nalaze se u jednom sustavu koji prikuplja i prezentira podatke za analizu
- Server-plus-sensors – u ovome načinu *SO* se ponaša kao distribuirana platforma u kojoj senzori prikupljaju podatke, a server ih agregira te omogućava udaljeni pristup i analizu podatka

Prvi način preporučljiv je novim korisnicima zbog jednostavne implementacije te u manjim mrežama u kojima je potrebno nadgledati samo jedan ili nekoliko segmenata koristeći jedan senzor. Nadalje, koraci instalaciju su dobro dokumentirani te ih je moguće naći na [1]. Također, jedan od dizajnerskih ciljeva prilikom razvoja *SO-a* bio je pojednostaviti instalaciju kako bi Windows administratori, koji se možda nisu susreli sa Linux sustavima, mogli lako i brzo osposobiti sustav. S obzirom na to detaljno objašnjavanje instalacije u ovome radu, biti će preskočeno.

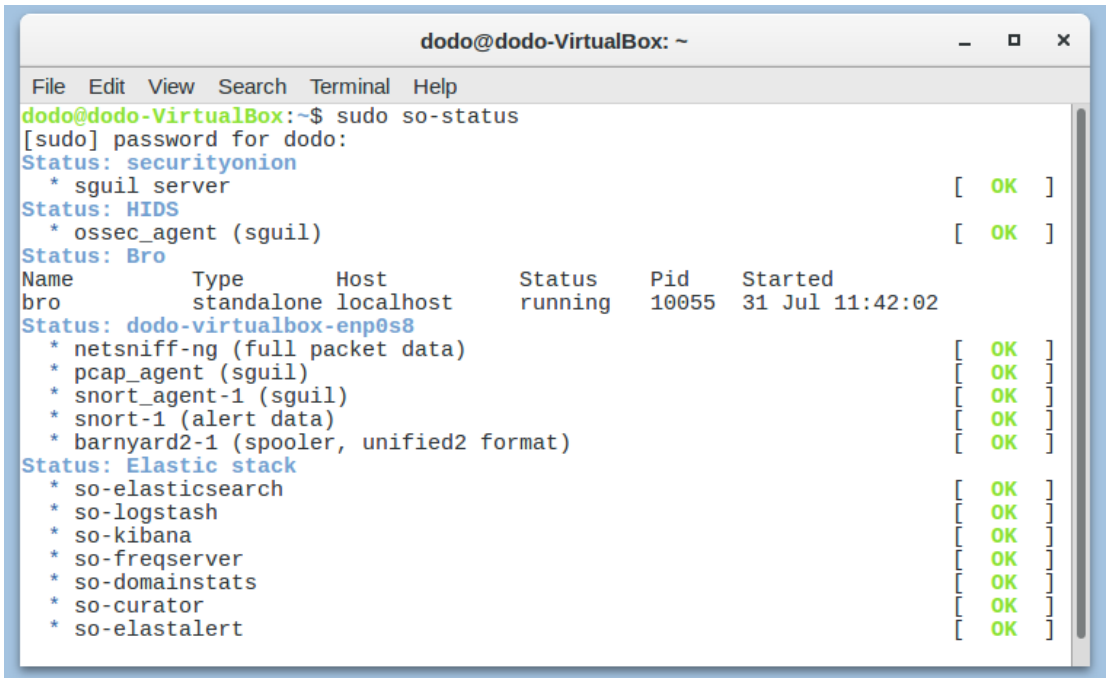
3.1 Virtualizacija

Za rad na *Securtiy Onionu* koristi će se *Oracle VM VirtualBox* koji omogućava virtualizaciju Linux sustava. Nakon dodavanja nove virtualne mašine potrebno je promijeniti određene postavke kao što je broj procesora i video memorija dodijeljena sustavu radi boljih performansi, *mountirati* .iso sliku kako bismo mogli provesti instalaciju *SO-a*. Nadalje, urediti mrežnu komponentu postavki. Odabiremo *Attached to: Bridged Adapter* kako bismo omogućili sučelju za nadgledanje (engl. *monitoring interface*) da dijeli mrežu našeg fizičkog uređaja (engl. *host*) kako bi mogao vidjeti mrežne aktivnosti koje se događaju na tome jednom sustavu. No ako je promatrati sve što je spojeno na *router* biti će potrebno postaviti preklopnik *(*engl. *switch)* koji će omogućiti da sučelje za nadgledanje prati sav promet koji dolazi u mrežu a ne samo onaj dio koji je specifično namijenjen za pojedini uređaj.

3.2. Otklanjanje poteškoća i održavanje

Poslije instalacije, najbolja praksa pri korištenju virtualnih mašina je kreiranje snimka (engl. *snapshot*), što omogućava vraćanje na dani trenutak u vremenu ako se dogodi nekakva greška. Iznimno bitna naredba za otklanjanje poteškoća je `$sudo so-status`, ova naredba obavit

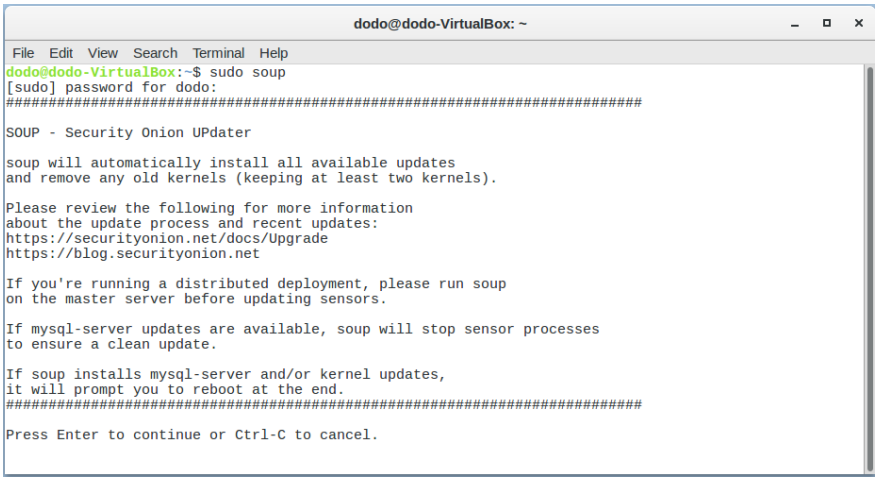
će provjere servisa i dati rezultate u terminal te ako išta u *SO-u* ne radi, kao na primjer *Squert* ne prikazuje uzbune, ovo bi trebao biti prvi korak pri otklanjanju poteškoća. Uključuje status mrežnog sučelja, iskorištenost diska i CPU-a, status ažuriranja IDS pravila i razne druge statuse vezane za performanse i „zdravlje“ *SO-a* . Ako se ikoja komponenta prikazuje kao neispravna (engl. *failed*) može nam olakšati otkrivanje problema. Zbog svega toga ovu naredbu bi trebalo usvojiti kao početni korak pri svakidašnjem nadziranju sustava. Slika 2.1 prikazuje da su sve funkcije *SO-a* pokrenute i da ispravno rade.



Slika 2.1 poželjan ispist $sudo so-status naredbe

3.2.1 Ažuriranje

NSM sustavi moraju biti regularno ažurirani kako ne bi došlo do slučaja da se pokreće kod koji ima određene ranjivosti. U tu svrhu kreirana je skripta „*soup*“, (skraćeno od Security Onion update), koja skida samo potrebne pakete za SO i jedini je način kojim bi trebali obavljati ažuriranja.

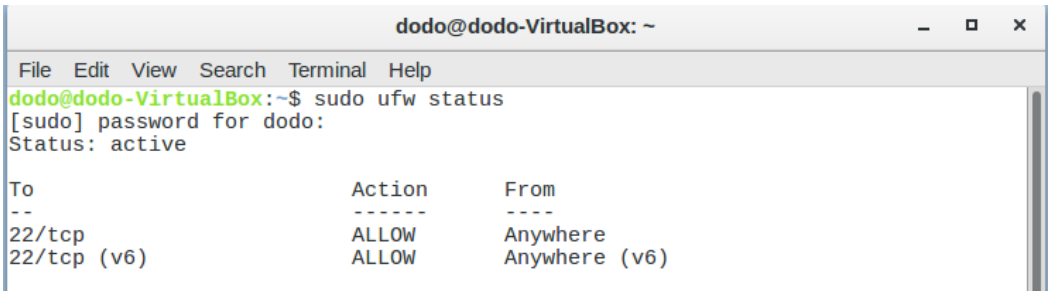


Slika 3.1 pokretanje soup naredbe

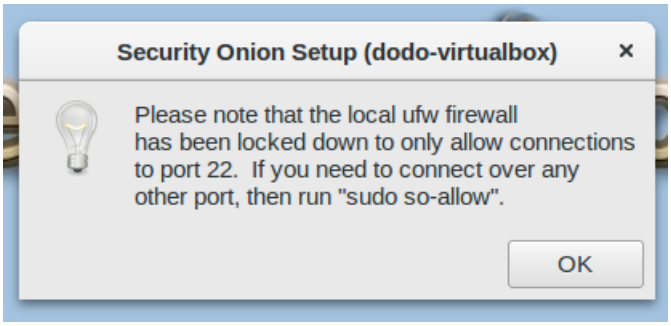
Skripta zahtjeva administrativne privilegije koje postižemo sa prijašnje navedenom naredbom `$sudo` koja zahtjeva lozinku odabranu tijekom instalacije. Ova naredba će biti često korištena iz tog razloga.

3.3.2 Ograničavanje pristupa SO-u

Po zadanom, SO dolazi sa lokalnim vatrozidom Linux sustava *iptables* koji pomaže promoviranju sigurnosti, a zadane postavke pristupa doznaju se načinom prikazanim slikom 4.1 Iz koje je vidljiv otvoren *port 22*, što je i očekivano s obzirom da *SO* obavijesti o tome prilikom instalacije (Slika 4.2)



Slika 4.1 status pristupnih portova SO-u



Slika 4.2 SO obavijest

Prema [4], Sintaksa za dodavanje dozvole ili ograničenja:

```
sudo ufw allow <port>/<optional: protocol>
sudo ufw deny <port>/<optional: protocol>
```

Postojeća pravila se brišu dodavanje naredbom `delete` ispred originalnog pravila, pa se pravilo:

```
sudo ufw deny 80/tcp
```

briše:

```
sudo ufw delete deny 80/tcp
```

3.3.3 Upravljanje pohranom podataka

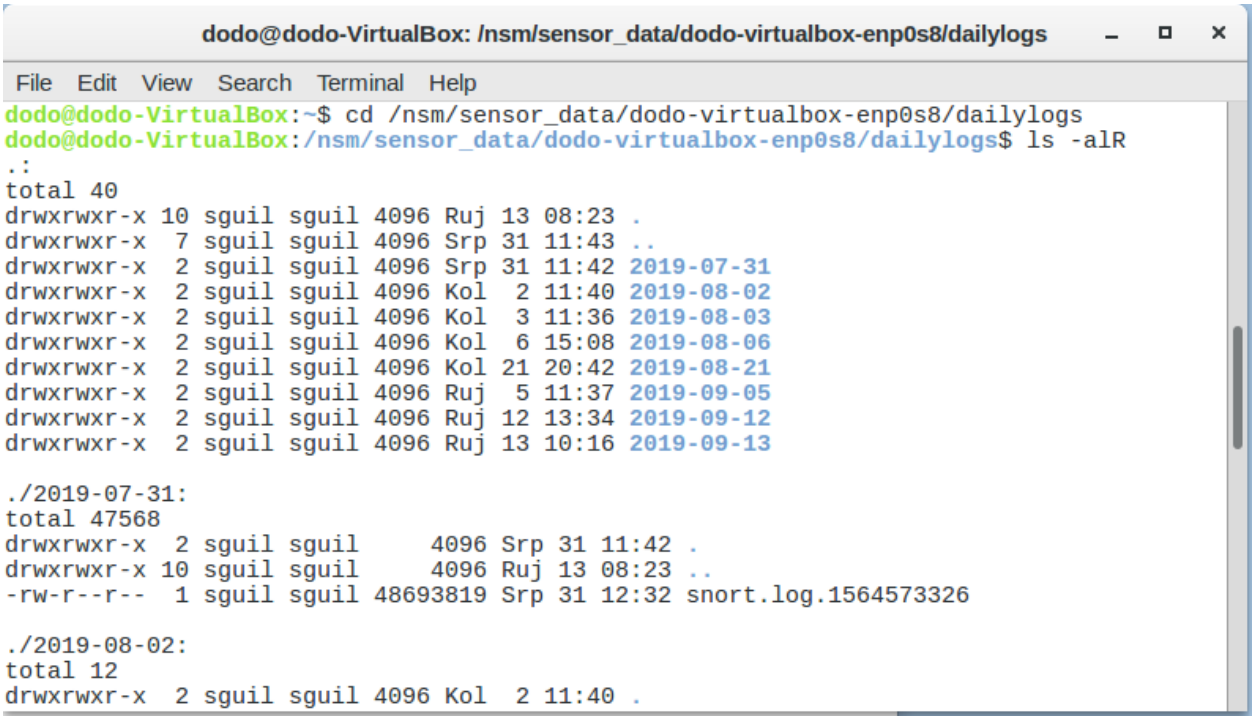
Spomenuto kako jedno od glavnih komponenti NSM-a je potpuno hvatanje paketa, što znači da se svi ti uhvaćeni paketi negdje moraju spremiti, što nadalje zahtjeva veliku količinu memorije i ogromne baze podataka koje je potrebno regularno održavati kako bi sustav radio optimalno te kako vrijedni podatci ne bi bili izgubljeni.

- `/nsm` direktorij pohranjuje zapise i podatke cijeli paketa
- `/var/lib/mysql` direktorij sadrži SO bazu podataka

SO sprema podatke cijelih paketa u obliku *.pcap* datoteka na lokaciji:

/nsm/sensor_data/<sensorname-interface>/dailylogs/YYYY-MM-DD

Prema slici 5.1 vidljivi su neki od cijelih paketa spremljenih tijekom korištenja *SO-a*.



Slika 5.1 direktoriji s podacima cijelih paketa

Prikazano vrijeme je trenutak kada je datoteka zadnji put izmijenjena, a vrijeme u *snort.log.<Unix timestamp>* je trenutak kreiranja izražen u sekundama proteklim od 1. siječnja 1970. godine. *SO* skripte se brinu o ovim podacima, tako što provjeravaju raspoloživost prostora tvrdog diska regularno, te kada premaši granicu od 90%, brišu se stariji podaci.

Za provjeru veličine *SO* baze podataka u */var/lib/mysql* moramo se prvo prijaviti na *MySQL* server te poslati upit prema [5] koji će nam vratiti veličinu baze podataka:

Tablica 2.1 SO baze podataka

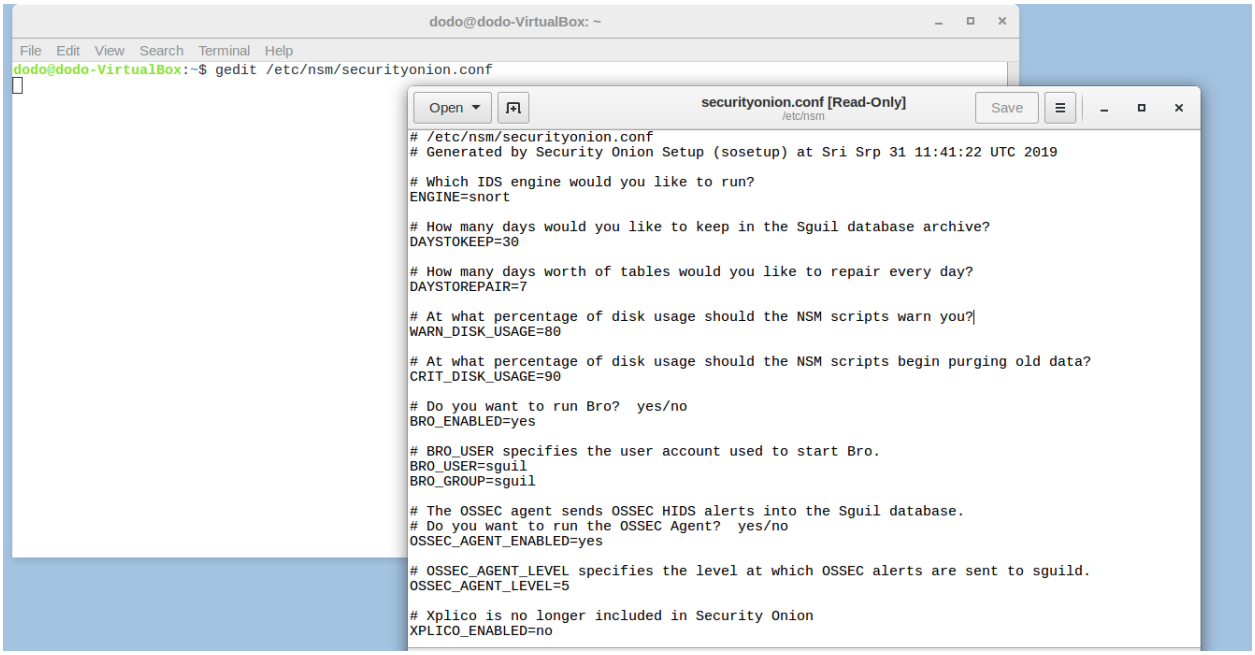
DBName	Data Size	Index Size	Total Size
sys	0.000 GB	0.000 GB	0.000 GB
securityonion_db	0.025 GB	0.031 GB	0.055 GB
All Databases	0.025 GB	0.031 GB	0.055 GB

3 rows in set (0,10 sec)

Tablica 2.1 prikazuje kako sav prostor zauzima `securityonion_db` koja je korištena od strane *Sguila* i njegovih komponenti, dok `sys` baza podataka korištena od strane *ELSA-e* ne zauzima prostor jer ti alati još nisu pokretani.

SO dolazi sa `sguil-db-purge` skriptom za upravljanje bazom podataka `securityonion_db`. Konfiguracijska datoteka se nalazi u `/etc/nsm/securityonion.conf` i sadrži varijablu

DAYSTOKEEP koja je po zadanom postavljena na 365, no moguća je promjena kao što prikazuje slika 5.2, u ovome primjeru, podaci se čuvaju 30 dana nakon čega se brišu.



Slika 5.2 promjena vrijednosti DAYSTOKEEP varijable

Iskorištenost diska, u ovom slučaju je 16%, a provjerava se dvjema naredbama prikazanim u Ispisu 2.1 Dok */nsm* direktorij zauzima 602MB od ukupnih 7.8GB kojeg zauzima cijela particija.

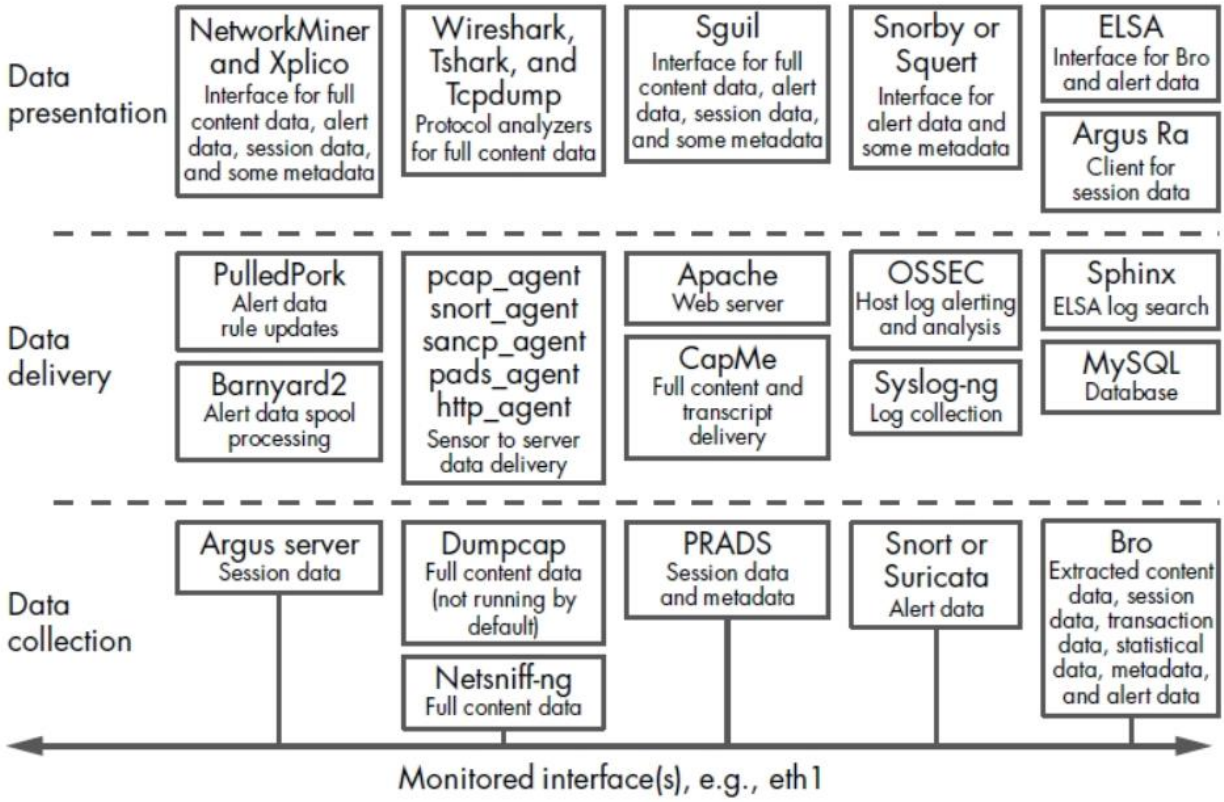
```
$ sudo df -h
[sudo] password for dodo:
Filesystem      Size  Used Avail Use% Mounted on
udev            2,4G     0  2,4G   0% /dev
tmpfs           484M  7,9M  476M   2% /run
/dev/sda1        52G  7,8G   41G  16% /
tmpfs           2,4G  272K  2,4G   1% /dev/shm
tmpfs           5,0M  4,0K  5,0M   1% /run/lock
tmpfs           2,4G     0  2,4G   0% /sys/fs/cgroup
tmpfs           484M     0  484M   0% /run/user/1001


$ sudo du -csh /nsm
602M  /nsm
602M  total
```

Ispis 2.1 Iskorištenost diska

4. ALATI

Nakon shvaćanja kako treba održavati SO sustav, da bi on radio optimalno vrijeme je upoznati se sa alatima i funkcionalnostima koje oni pružaju. Alati uključeni u SO su mnogobrojni te je kompletan popis moguće naći na [6]. Nadalje, moguće ih je podijeliti u tri skupine: sakupljanje podataka, prezentacija podataka te dostava podatak (Slika 6.1)



Slika 6.1 podjela SO alata [2, str 116]

4.1 SO alati prezentacije podataka

Preko ovih alata stvara se uvid u promet koji teče mrežom te mogućnost analize paketa, stoga se još nazivaju i alati za analizu. Također, možemo ih podijeliti na osnovi korisničkog sučelja koje pružaju: alati sa tekstualnim i alati sa grafičkim sučeljem. Dodatno SO uključuje i NSM konzole za prikaz podataka. Alati za analizu paketa čitaju promet kako on teče kroz žicu („uživo“) ili iz *.pcap* datoteka. Od alata sa tekstualnim sučeljem, na raspolaganju su: *Tcpdump, Tshark, Argus Ra.*Grafičko sučelje pružaju alati: *Wireshark, Xplico i NetworkMiner*. Ovi alati nisu dizajnirani specifično za NSM primjenu, no omogućuju analitičarima bolje razumijevanje pojedinih paketa, grupiranje pakete u sesije i razne druge funkcionalnosti koje su ključne za

razumijevanje mrežnog prometa. S druge strane, NSM konzole dizajnirane su specifično za NSM orijentiranu primjenu, a uključuju: *Sguil, Squert, Snorby i ELSA*.

4.2 SO alati kolekcije podataka

Ova skupina alata zadužena je za skupljanje i generiranje podataka koji bivaju analizirani i vizualizirani kroz alate prezentacije. Njima pripadaju: *Netsniff-ng, Snort, Suricata i Bro*. *Netsniff-ng* zadužen je za potpuno hvatanje paketa te njihovo spremanje u *.pcap* formatu. *Snort i Suricata* predstavljaju *NIDS*-ove te je mogućnost odabrati između te dvije opcije prilikom instalacije. *Bro* nadgleda i interpretira promet koji je generiran. Sve ove komponente generiraju i više nego dovoljno podataka koji se analiziraju prijašnje navedenom skupinom alata.
U trenutno instalaciji korišten je *Snort,* mrežni sustav za otkrivanje upada temeljen na uzorcima/potpisima. Pravila ovog alata održava *PulledPork* koji svaki dan ažurira bazu uzoraka, no uz njega postoji mogućnost prilagođavanja i pisanja vlastitih pravila koja se nalaze u */etc/nsm/rules/local.rules*

4.3 SO alati za dostavu podataka

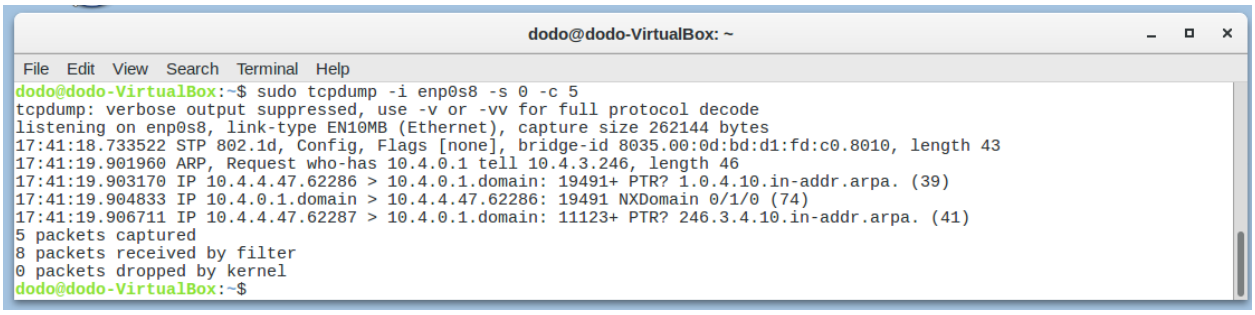
Alati za dostavu povezuju prijašnje dvije skupine, odnosno omogućavaju njihove funkcionalnosti. Alati kao: *PulledPork, Barnyard2 i CapMe* upravljaju IDS pravilima. Nadalje pivotiranje iz jednog alata u drugi omogućava prikaza istih podataka, različitim alatima dozvoljavaju formiranje šireg konteksta oko generiranog prometa te lakše zaključivanje i donošenje odluka. Također uključeni su i alati koji omogućavaju određene *HIDS* analize kao što su: *OSSEC* i *Syslog-ng* te *Wazuh.*

4.4 Tcpdump

Prema [3], *Tcpdump* je alat za hvatanje paketa s tekstualnim sučeljem. Baziran je na UNIX-u, što znači da dolazi ugrađen sa većinom UNIX-baziranih sustava. Nadalje, ima mogućnost hvatanja paketa kako oni teku kroz mrežu („uživo“) ili ako su spremljeni u *.pcap* datoteku. Iz tih razloga vrlo je korisno upoznati se s ovim alatom, jer ga je moguće koristiti na skoro svakoj mašini koja ima pristup terminalu.
Za pokretanje hvatanja paketa „uživo“ koristi se sljedeća sintaksu sa odgovarajućim parametrima za pojedine opcije:

```
$ sudo tcpdump –i <interface> -s <snaplet> -c <count>
```

Opcija *–i* određuje mrežno sučelje na koje se „izbacuje“ (engl. dumpa) promet, opcija *–s* određuje koliko bajtova će se uhvati od svakog paketa (odabirom *–s 0*, bit će uhvaćeni cijeli paketi), opcija *–c* određuje koliko paketa želimo uhvatiti, stoga ako je izostavljena *tcpdump* će hvatati pakete dok se ne zaustavi sa *ctlr+c*.



Slika 7.1 tcpdump ispis

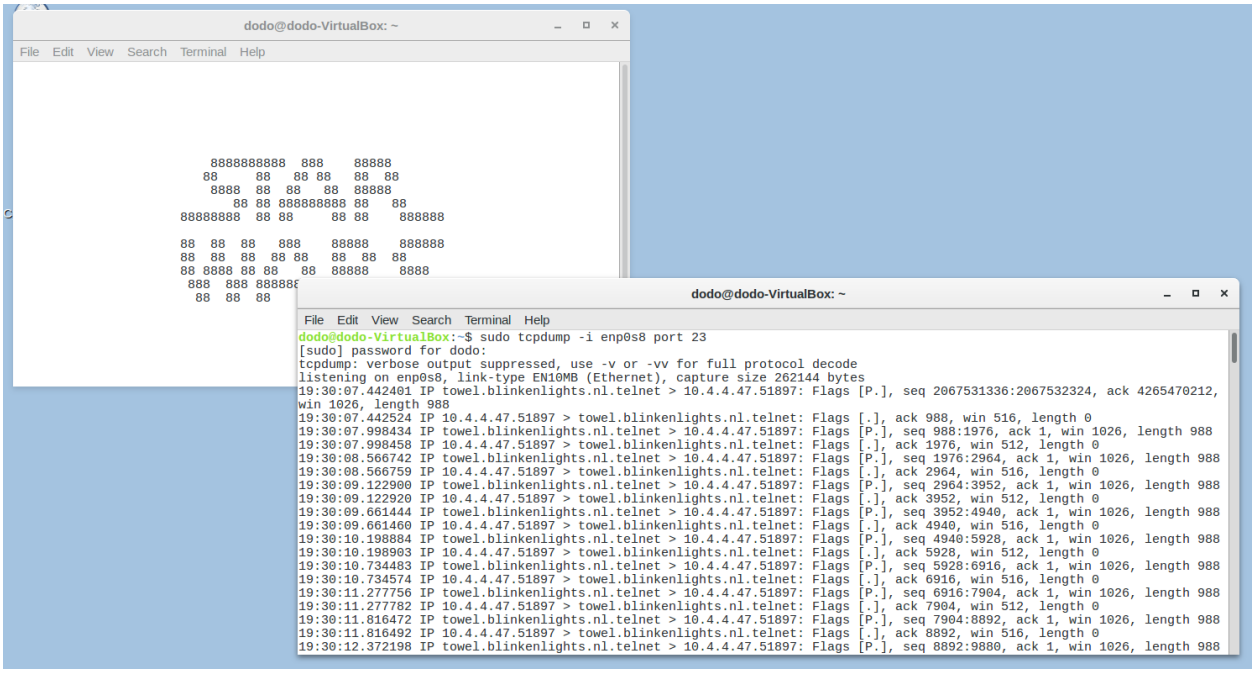
Slika 7.1 prikazuje 5 paketa, uhvaćenih na *enp0s8* sučelju, što je ustvari „osluškivano“ sučelje (engl. *sniffing interface)* odabrano za nadgledanje tijekom instalacije. U ispisu se može uočiti da svaki paket dolazi sa vremenskom oznakom kada je uhvaćen. Analizom uhvaćenih paketa da se razlučiti postavljanje logičke topologije mreže kroz *STP* protokol, razašiljanje *ARP* protokola kako bi se saznala fizička adresa te komunikacija sa *DNS* poslužiteljem kako bi se razriješile adrese. Dodavanjem opcije *–w <naziv_datoteke>* moguće je pakete spremiti u odabranu datoteku, dok za čitanje iz datoteke koristimo *–r <naziv_datoteke>*.

Nakon uhvaćenih svih paketa koji prolaze kroz mrežu, te analize istih zaključuje se kako većina toga nije zanimljivo. Potrebno je koristiti filtre kako bi se zanemario „nebitni“ promet, a spremilo nešto zanimljivo. Filtrirat će se *telnet* konekcije uspostavljena sa sljedećim serverom:

```
$ telnet towel.blinkenlights.nl
```

Naredba pokreće *Star Wars, episode IV* film preko *telnet* konekcije. Poznato je da *telnet* server po zadanom sluša na port 23 te je dovoljno koristit sljedeće kako bi se zabilježili samo oni paketi koji se razmjenjuju na ovoj konekciji.

```
$ sudo tcpdump –i enp0s8 port 23
```

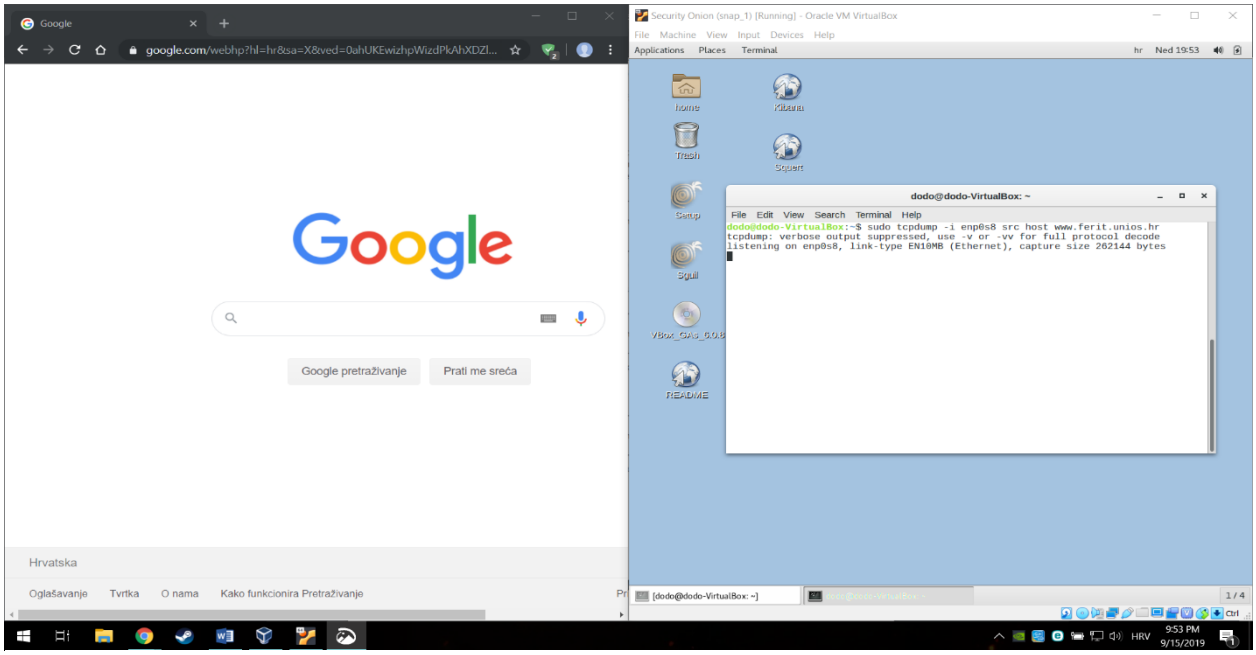


Slika 7.2 filtri uz tcpdump

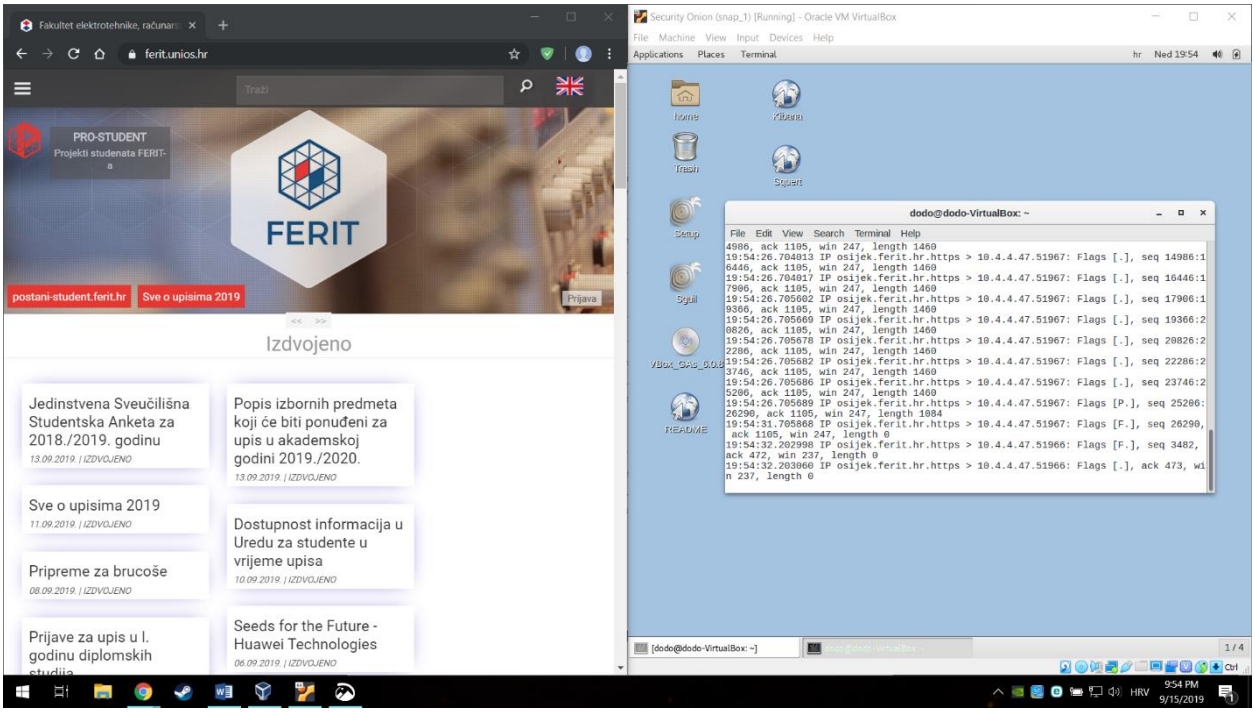
Slika 7.2 prikazuje ispis *tcpdump-a* uz korištenje filtra, te je jedini promet zabilježen samo onaj između sučelja *enp0s8* i *telnet* servera.

Ako je pak potrebno uhvatiti samo one pakete koji dolaze sa specifičnog odredišta koristi se sljedeća sintaksa:

```
$ sudo tcpdump -i enp0s8 src host www.ferit.unios.hr
```



Slika 7.3 src host filtar



Slika 7.4 src host filtar

Na slikama 7.3 i 7.4 može se vidjeti da se ne bilježi nikakav promet sve dok se ne posjeti stranica navedena parametrom *src host* filtra.

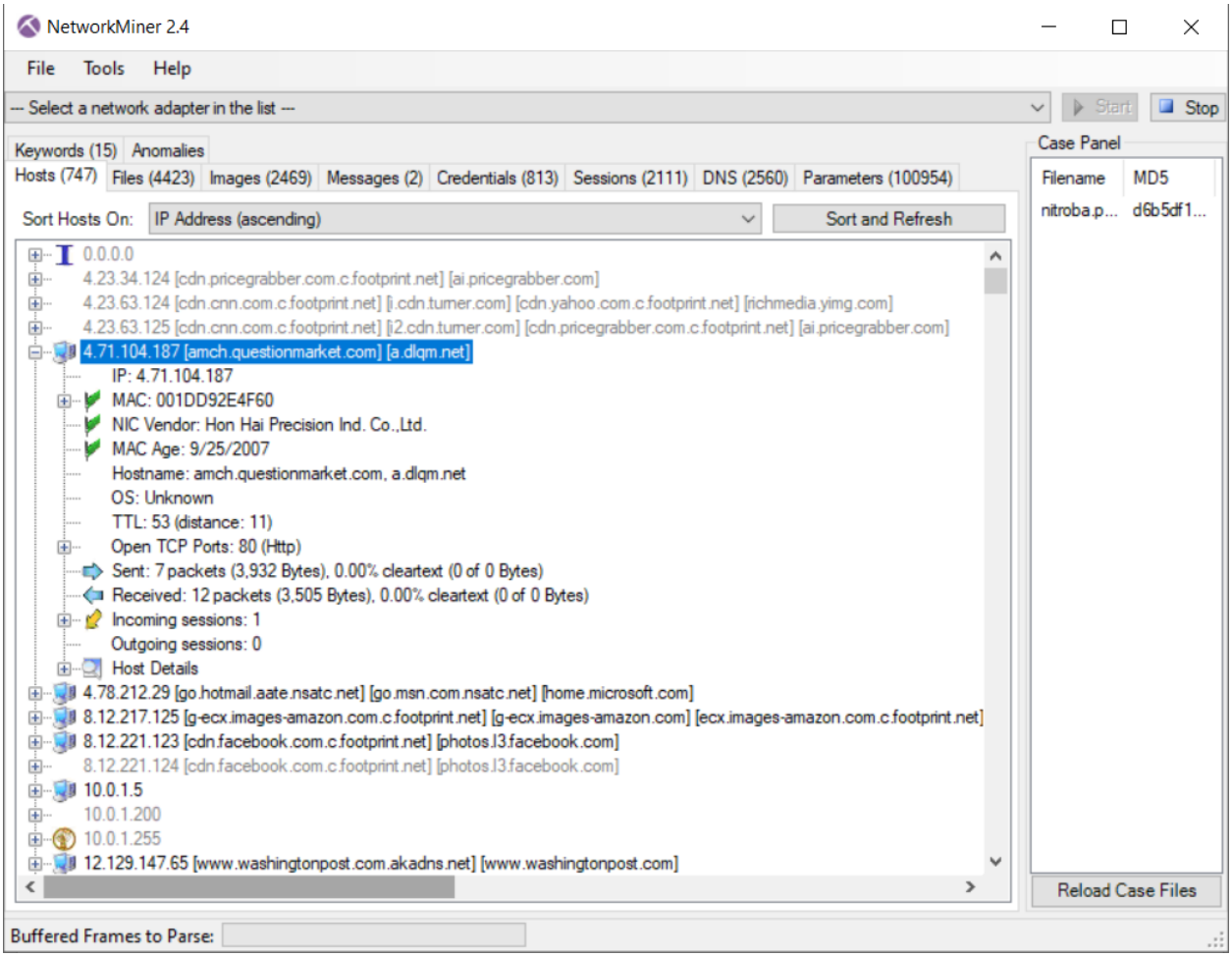
Analogno ovim primjerima mogu se koristit razni drugi filtri kako bi se ograničilo hvatanje samo onog prometa koji je iz nekog razloga potreba, zanimljiv, ili pak drugačiji. Najbolji prijatelj pri pisanju filtara jest svakako `&man pcap-filter`, gdje je moguće pronaći sve filtre koji bi ikad mogli zatrebati.

U konačnici, *tcpdump* daje brz i jednostavan način hvatanja te filtriranja mrežnog prometa kako bi se izvukao samo onaj dio koji nam je najinteresantniji. Dodatno, po potrebi spremanje tog promet u *.pcap* formatu te pivotiranje u neki drugi alat, kako bi se dodatno analizirao.

4.5 NetworkMiner

NetworkMiner jest alat mrežne forenzičke analize (*NFAT)* koji spada u skupinu alata prezentacije podataka, sa grafičkim sučeljem. *NetowrkMiner* izvlači datoteke, e-mailove, certifikate, sve IP adrese i razne druge detalje što znatno olakšava posao te štedi vrijeme pri analiziranju prometa. Grafičko sučelje *NetworkMinera* prikazano je slikom 8.1, sadrži niz kartica sa korisnim podacima. Na primjer, *Host* kartica prikazuje sve IP adrese koje je *NM* zapazio tijekom rekonstrukcije *.pcap* datoteke, također pruža metapodatke o svakoj pojedinoj adresi, kao što su MAC adrese, odlazne i dolazne sesije i još mnogo toga. Još jedna korisna funkcionalnost koju treba spomenuti su ključne

riječi (engl. *Keywords*) koje se unose u *string* ili *hex* formatu te na taj način pretražujemo učitane podatke.



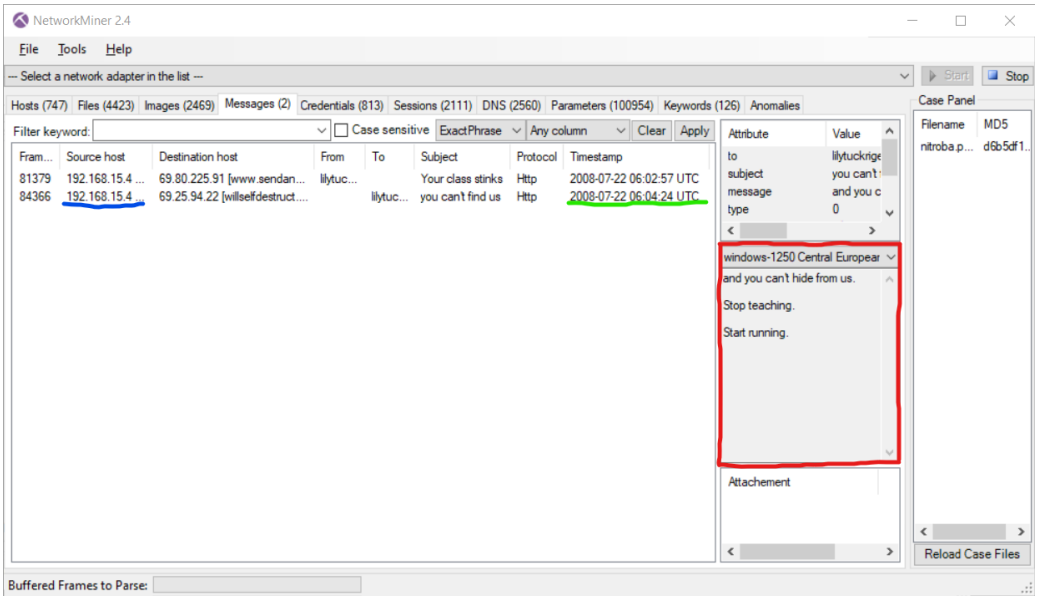
Slika 8.1 NetowrkMiner GUI

Za razumijevanje ovog alata primijenit ćemo mrežni promet dostupan na D*igitalCorpora.org*, što je stranica sa kolekcijom digitalnih dokaza kao što su podaci mobilnih uređaja, mrežni paketi, slike diskova i razne datoteke. Sve ovo je dostupno i slobodno za korištenje analitičarima i studentima u svrhu učenja tehnika i alata koji obrađuju navedene podatke.

Scenarij za koji će biti vezani prikazani podaci jest *Nitroba University Harasment Scenario* prema [7], u kojem profesorica kemije dobiva prijeteće *e-mailove* od anonimnog studenta. Nakon prijave ovog događaja administraciji fakulteta, počinje se pratiti mrežni promet te se čeka ponovni napad, koji se događa nekoliko dana poslije.

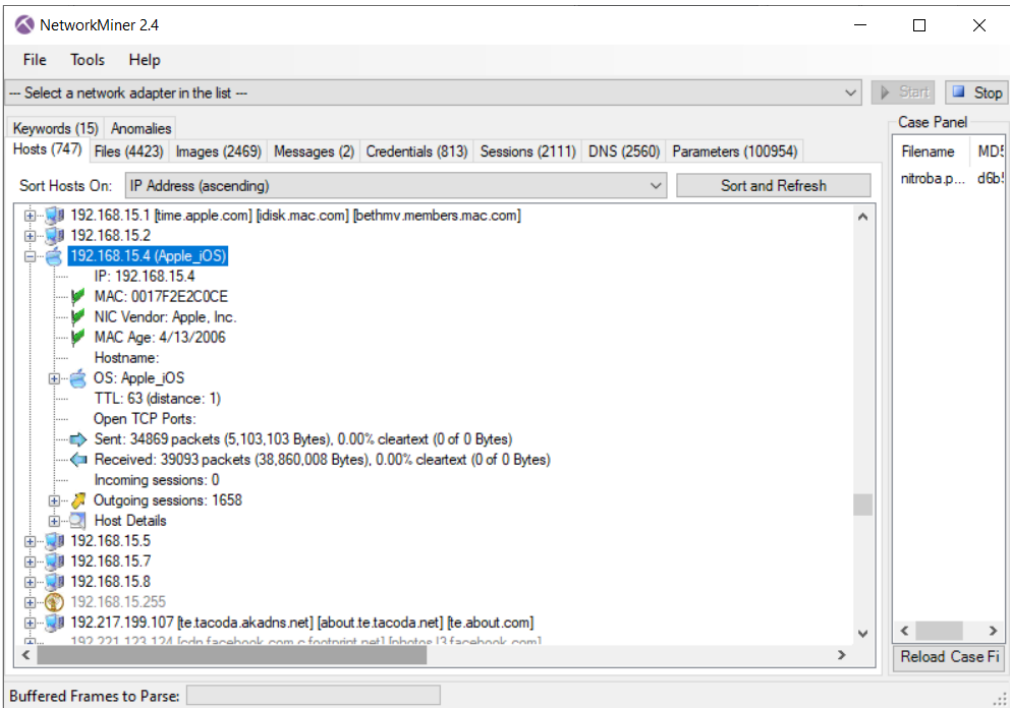
Nakon učitavanja *nitroba.pcap* datoteke, cilj je otkriti tko šalje anonimne, uznemiravajuće e-mailove. Uočljiva je kartica *Messages*, koja daje uvid e-mailova, izlučenih od strane

NetworkMinera. Slika 8.2 prikazuje poruku (crveno), izvorišnu adresu (plavo) i vrijeme slanja (zeleno).



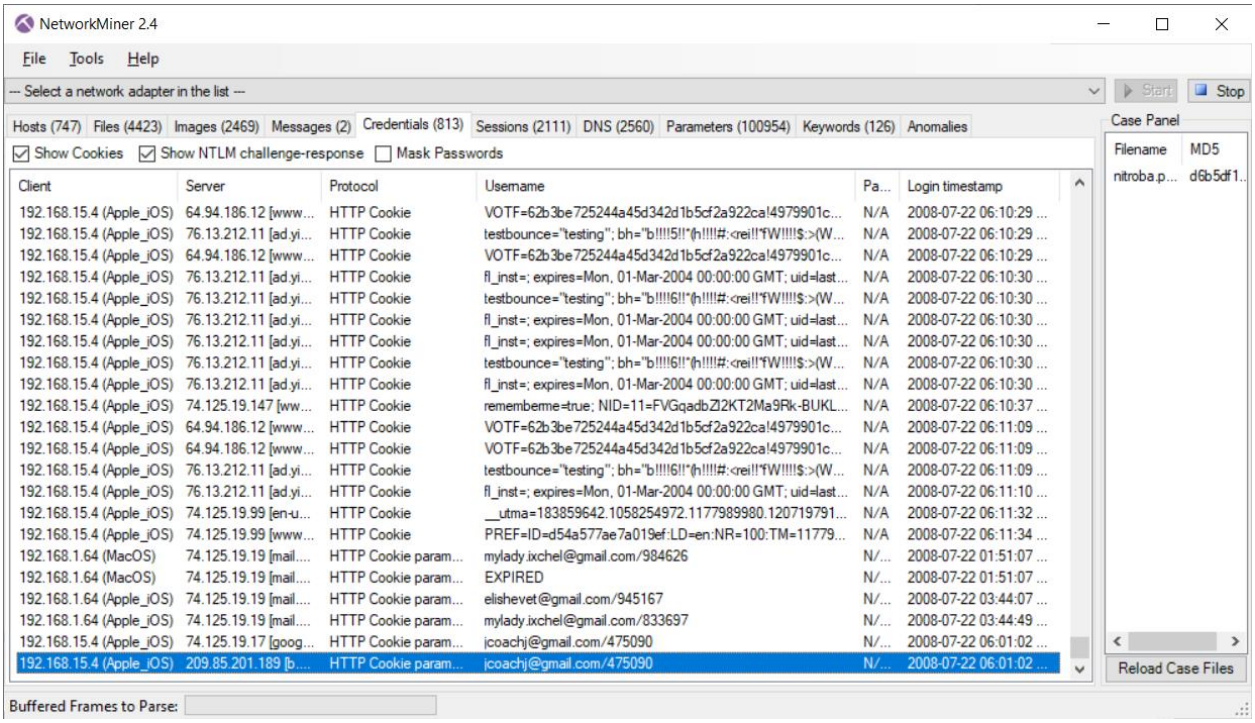
Slika 8.2 pregled zabilježenih e-mailova

Doznata IP adresa pošiljatelja (192.168.15.4) može se potražiti u *Host* kartici. Na slici 8.3 prikazano je kako adresa pripada *Apple* uređaju



Slika 8.3 IP podaci i meta-podaci pošiljatelja

Još jedna od zanimljivih funkcija *NetworkMinera* je izdvajanje korisničkog imena i zaporke (engl. *credentials)* spremljenih u odgovarajućoj kartici , prikazanoj na slici 8.4. Ovdje se može pronaći *HTTP Cookie parametar* sa korisničkim imenom jcoachj@gmail.com, čija je prijava zabilježena na osumnjičenoj IP adresi, nekoliko minuta prije nego što su s iste poslani anonimni uznemiravajući e-mailovi. S obzirom da je *Johhny Coach* jedan od studenata koji pohađaju nastavu profesorice kojoj su upućeni e-mailovi, upravo on postaje glavni osumnjičen.



Slika 8.4 saznavanje korisničkog imena

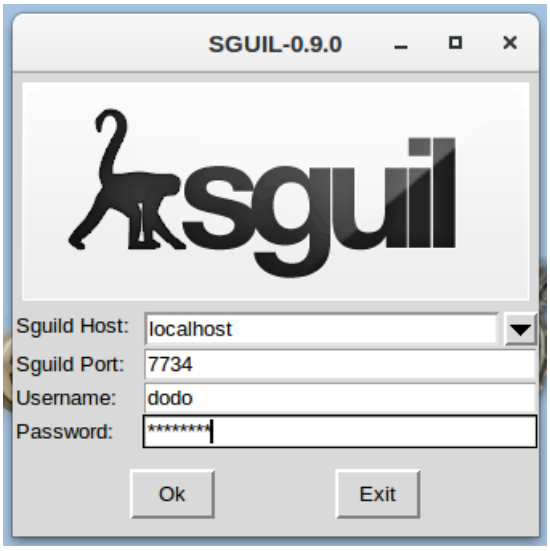
Zaključno s ovom pretpostavkom, pokazane su neke od glavnih funkcija *NetworkMinera,* jednostavnog i poprilično intuitivnog NFA alata. Korištenje ovog alata štedi vrijeme analitičara, jer na uvid daje sve moguće datoteke, slike, poruke, akreditacija, već izlučene i sortirane u zasebne kartice te preostaje jedino proći kroz sav taj promet i napraviti smisao od njega.

4.6 Sguil

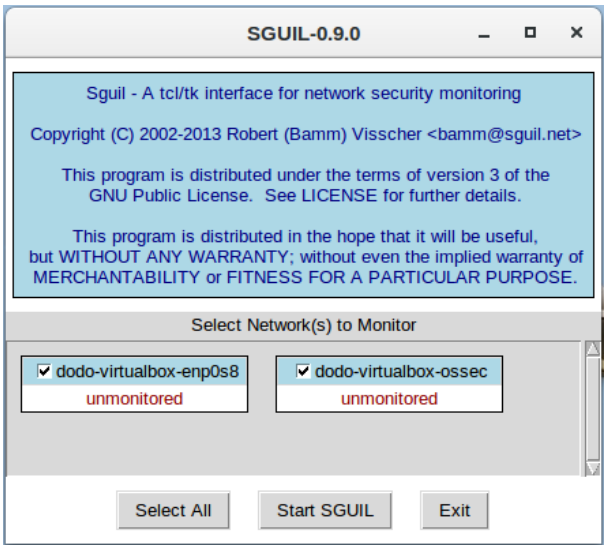
Za razliku od prijašnje navedenih alata, *Sguil* je prezentirajući alat koji je razvijen od strane analitičara mrežnog prometa za analitičare mrežnog prometa. Dok se prijašnji alati mogu smatrati „proizvođačima“ NSM podataka, na *Sguil* treba gledati kao „potrošača“ tih podataka. Pisan je u Tcl/Tk i može se pokrenuti isključivo „uživo“, odnosno da prati promet koji se trenutno odvija na

mreži. Pokreće se dvostrukim klikom na ikonu koja se nalazi na radnoj površini SO-a, nakon čega unosimo korisničko ime i lozinku kao što je prikazano na slici 9.1.

Nakon prijave odabiremo mrežna sučelja koja će se „osluškivati“, prikazano slikom 9.2.

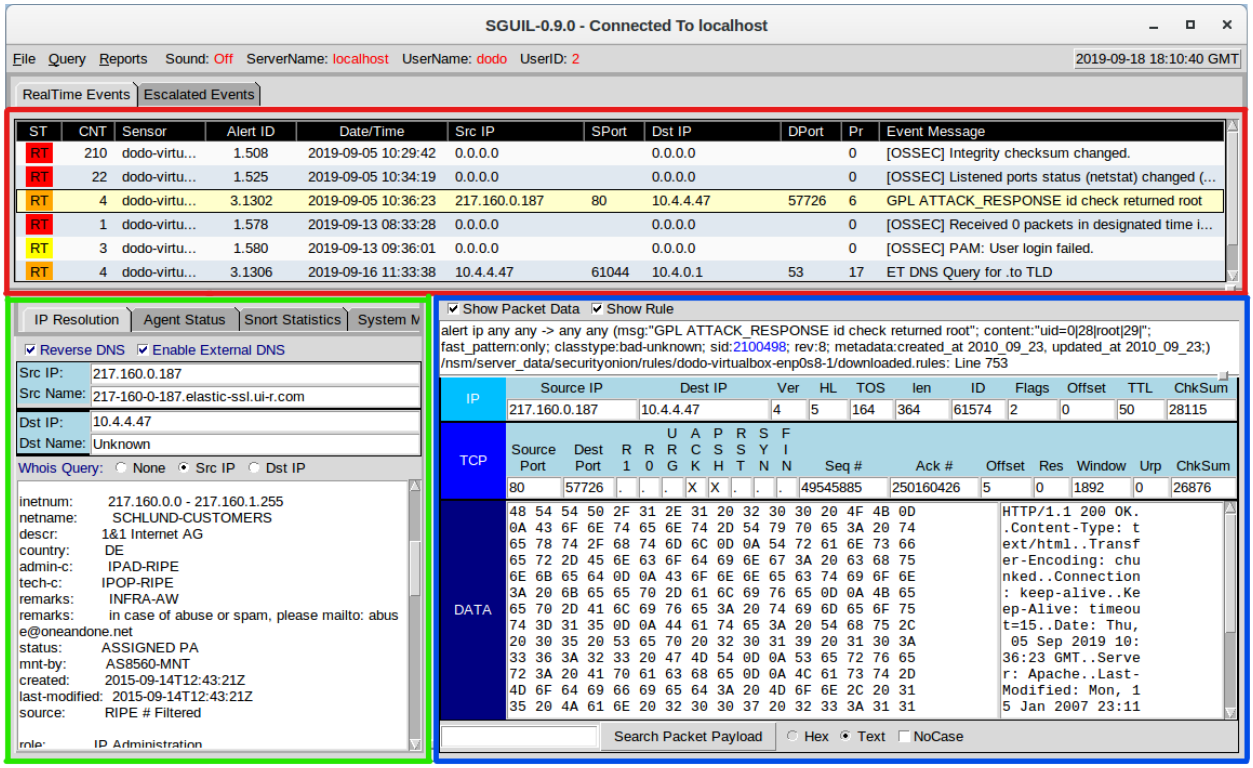


Slika 9.1 Sguil prijava



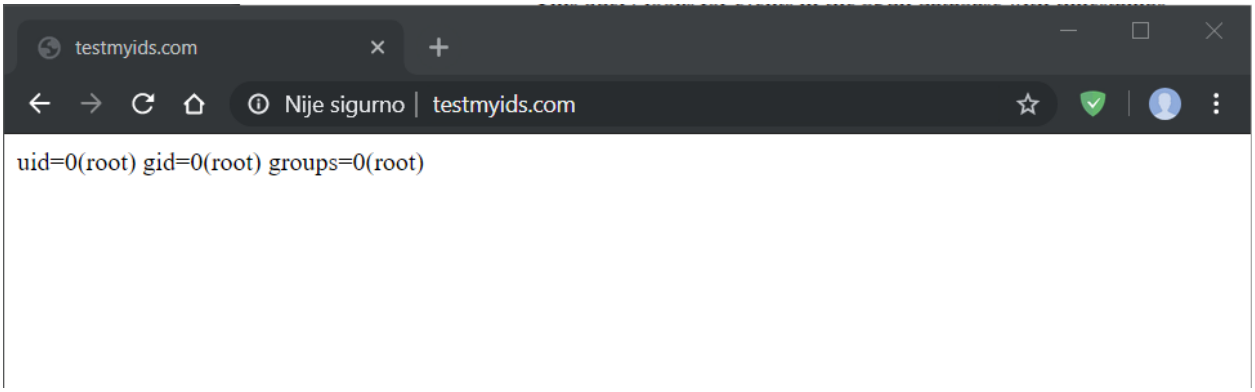
Slika 9.2 odabir mrežnih sučelja

Nakon ispravne prijave i odabira sučelja, prikazuje se *Sguil* konzola kao na slici 9.3 gdje se mogu vidjeti generirane uzbune (crveno), IDS pravilo i podaci paketa (plavo) te pojedinosti izvorišne i odredišne adrese (zeleno).



Slika 6.3 Sguil konzola s podacima

Među generiranim uzbuna vidljivo je nekoliko različitih vrsta:
OSSEC – uzbune vezane za HIDS kao što su otvaranje novog porta prikazano u potpoglavlju 3.3.2, unošenje krive lozinke prilikom prijave u sustav i slično
ET i *GPL* – uzbune vezane za NIDS koje generiraju *Snort* ili *Suricata* kada promet koji nadgledaju prekrši jedno od njihovih pravila. Na primjer, odabrana uzbuna na slici 16.3 prikazuje *GPL ATTACK_RESPONSE id check, returned root* koja se generira kada povrat UNIX naredbe *id* ima vrijednost vezanu za *root* korisnika. Ova uzbuna posljedica je posjećivanja *testmyids.com* stranice kako bi se utvrdilo radi li IDS ispravno. Naime stranica jednostavno ispiše tekst prikazan na slici 9.4 što izazove uzbunu od strane IDS jer ne razlikuje test od stvarnog sigurnosnog problema.



Slika 9.4 testmyids.com

4.6.1 Port scan

Skeniranje *portova (*engl. *port scan*) je jedan od prvih napada koje provalnik pokuša. Taj napada jednostavno daje do znanja provalniku koje aplikacije i usluge ciljani uređaj ima pokrenute odnosno koje su slabe točke sustava tako što šalje zahtjev za uspostavu konekcije i bilježi koji *portovi* odgovaraju i kako. *Port* može odgovoriti na tri načina:

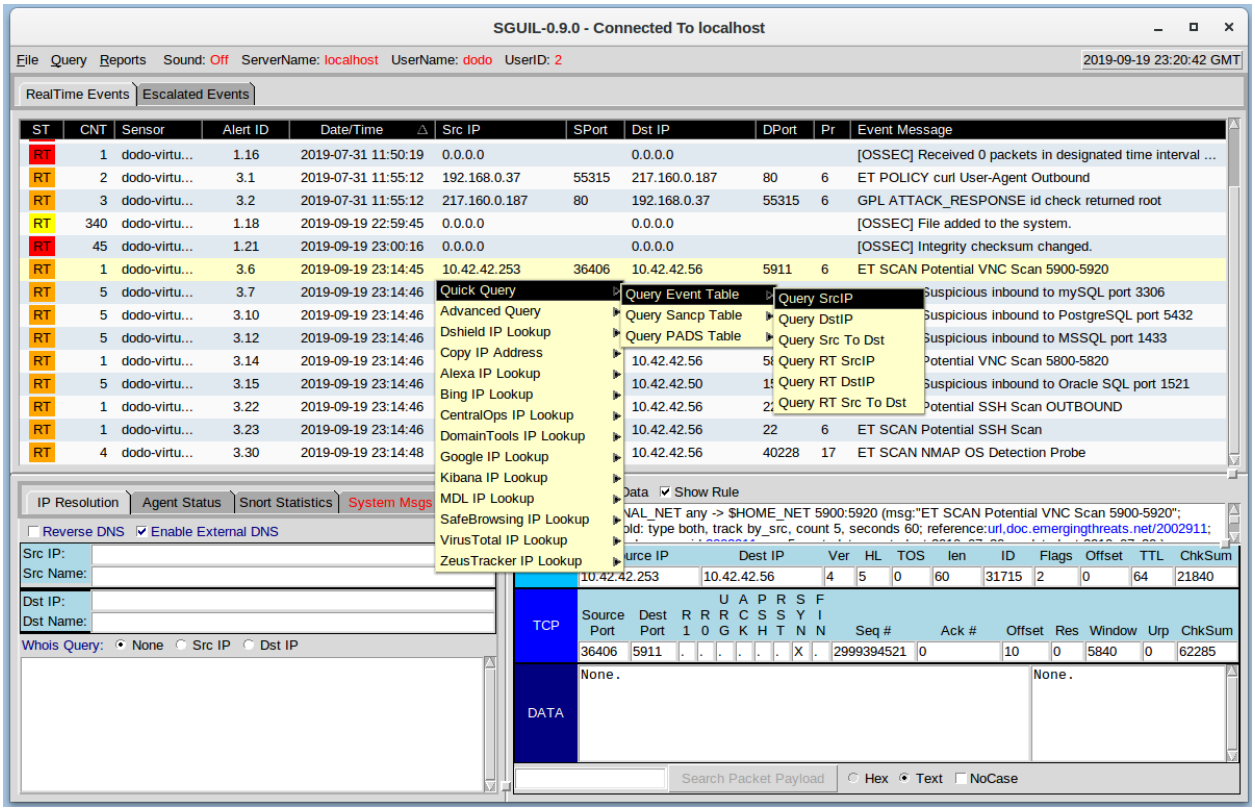
- OTVOREN *(engl. Open)* – odgovara na zahtjev za konekciju te daje provalniku do znanja da postoji uređaj na drugoj strani IP adrese koju je skenirao
- *ZATVOREN (engl. Closed)* – također će odgovoriti, no odbiti će zahtjev za konekciju, što i dalje daje provalniku do znanja da postoji uređaj
- NEMA ODGOVORA – ako *port* ne odgovori znači da je blokiran od strane vatrozida

Pravovremeno otkrivanje ovog napada vrlo je korisno, jer kao što je navedeno to je obično prvi napad koji se događa prije provale. U svrhu generiranja uzbuna vezanih za *port scanove* koristit će se primjer datoteka koja dolazi integrirana sa SO-om a nalazi se u a izvorno sa [8], */upt/samples/markofu/netforensics_evidence04.pcap* . Pomoću naredbe prikazane u ispisu 1.1, promet zabilježen u toj datoteci ponovno prenosimo na sučelje *enp0s8* brzinom od 10mbps.

```
&sudo tcpreplay -ienp0s8 -M10/upt/samples/markofu/netforensics_evidence04.pcap
```

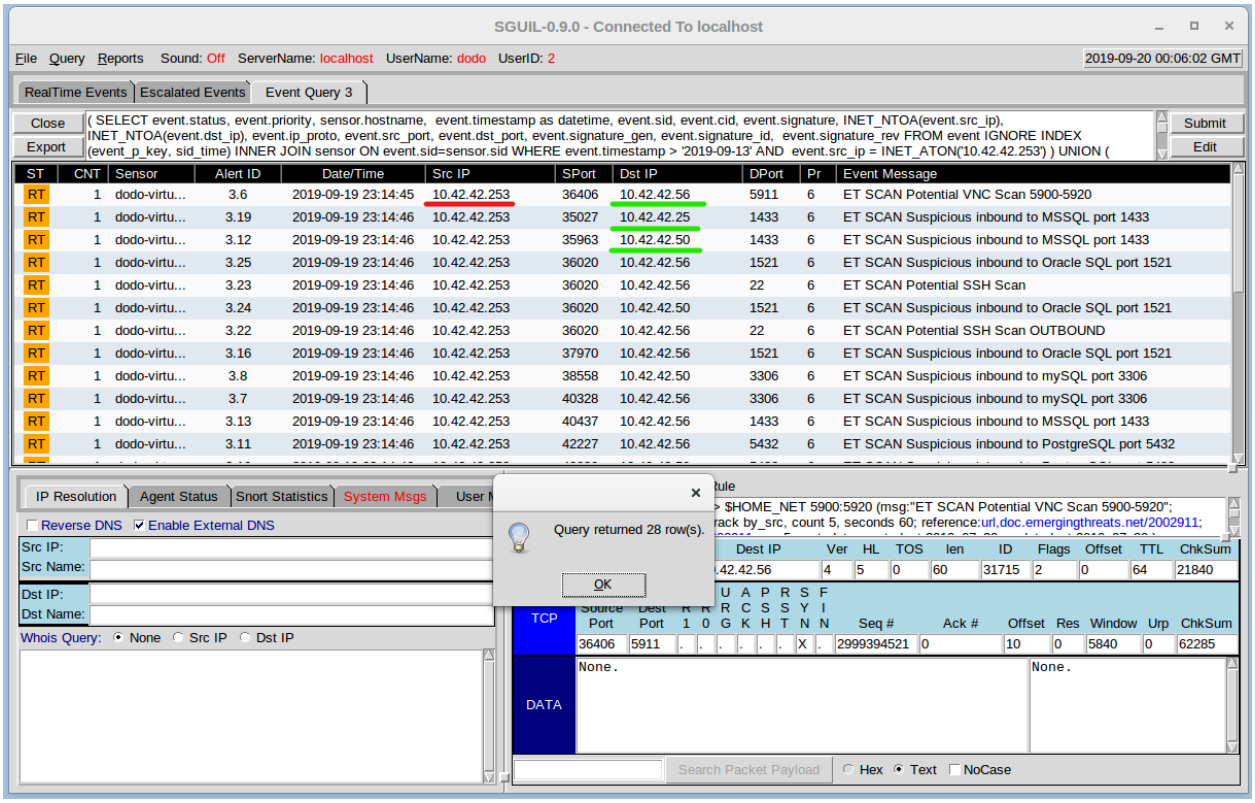
Ispis 3.1 tcpreplay navedenog .pcap filea

U *Sguil* konzoli pojavljuju se određeni broj uzbuna, koje se mogu grupirati ovisno o izvorišnoj adresi, kako bi se moglo fokusirati samo na njih. Navedeno je prikazano na slici 9.5



Slika 9.5 Stvaranje query event table prema izvorišnoj adresi

Ukupno postoji 28 uzbuna koje su zabilježene od strane *Sguila*, te je moguće odrediti kako *port scanove* izvodi IP adresa 10.42.42.253, označena crveno na slici 9.7, dok skenirane bivaju IP adrese 10.42.42.25, 10.42.42.50 i 10.42.42.56, označeni zeleno. Dodatno, iz tih informacija moguće je zaključiti kako provalnik izvodi *port scan* lokalno.



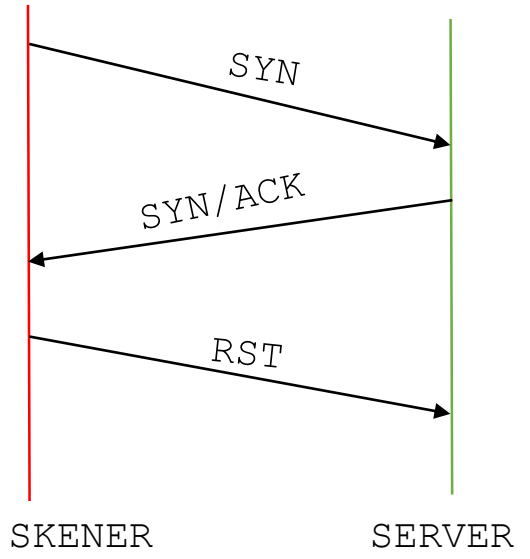
Slika 9.7 Query event table izvorišne adrese

Ako su potrebne dodatne informacije kako bi se utvrdile okolnosti ovih uzbuna moguće je pivotirati u razne druge alate kako bi se prikazali podaci potpunih paketa. To se postiže desnim klikom na *Alert ID*, te odabirom alata kao na primjer *Wireshark*.

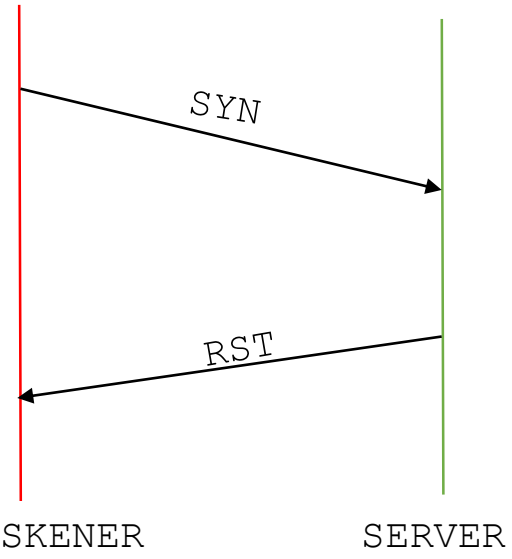
Jedan primjer okolnosti koje se mogu ovdje utvrditi jest vrsta *port scana* koja je korištena. Naime postoji više tehnika *port scenova* od kojih su najčešće:

4.6.2 TCP Syn scan

Šalje se TCP paket sa postavljenom Syn (0x002) zastavicom za uspostavu veze. Ako je *port* otvoren server odgovara sa TCP paketom sa postavljenim Syn/Ack (0x012) zastavicom, nakon čega skener prekida vezu sa Rst (0x004) zastavicom. U slučaju da je *port* zatvoren, server odgovara sa Rst/Ack (0x014) zastavicom. Ovaj *scan* se još naziva polu-konekcija jer ne ispunjava sinkronizaciju u tri koraka (Syn-Syn/Ack-Ack). Slike 10.1 i 10.2 prikazuju ovu interakciju odgovarajuće



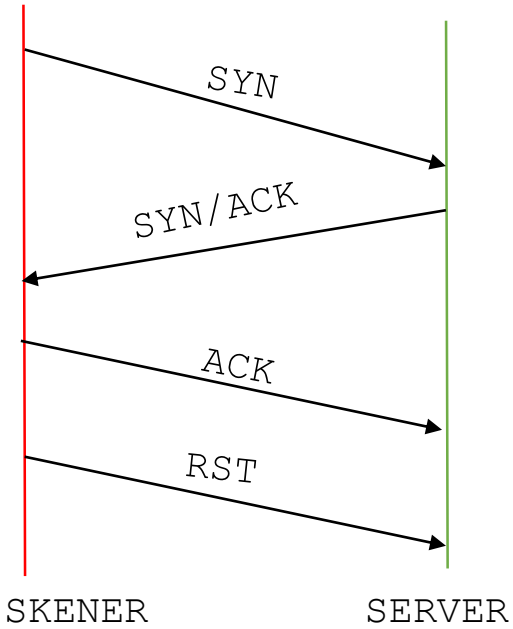
Slika 10.1 TCP Syn scan otvorenog porta



Slika 10.2 TCP Syn scan zatvorenog porta

4.6.3 TCP Connect scan

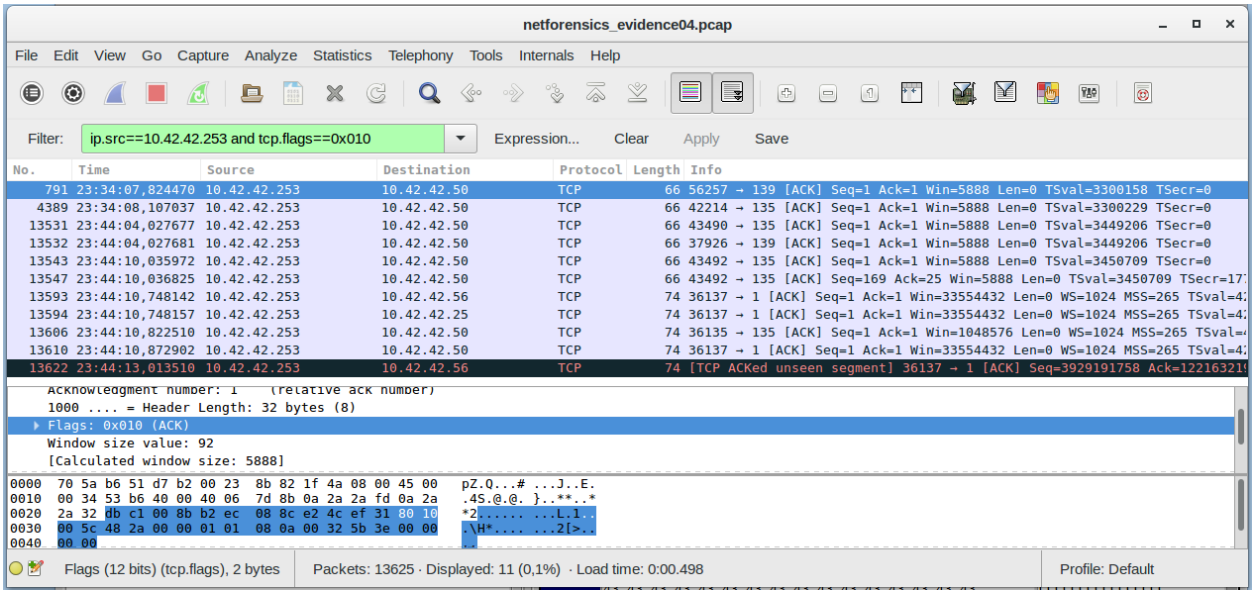
Ovaj scan počinje isto kao i prijašnji, slanjem TCP Syn (0x002) zastavice, nakon čega server, ako je otvoren, odgovara Syn/Ack (0x012), zatim skener uspostavlja vezu slanjem Ack (0x010) kako bi potvrdio primanje Syn/Ack paketa poslanog od strane servera. Odmah nakon toga terminira konekciju slanjem Rst (0x004). Ako je *port* zatvoren događa se jednaka komunikacija kao i u prijašnjem slučaju.



Slika 10.3 TCP Connect scan

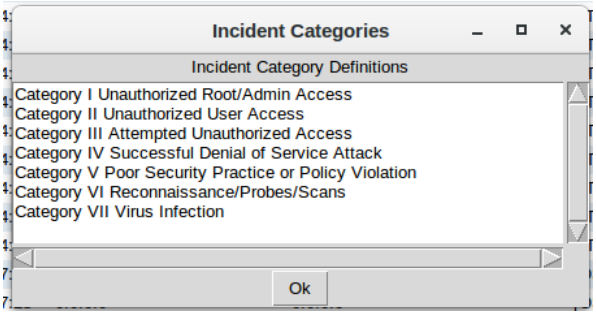
Shvaćajući kako pojedini *scan* funkcionira, moguće je u *Wiresharku* postaviti sljedeći filter kako bi se provjerilo da li skener uspostavlja konekciju te odredilo koristi li se *TCP Connect scan ili TCP Syn scan.*

```
ip.src==10.42.42.253 and tcp.flags==0x010
```



Slika11.1 Wireshark filter

Na slici 11.1 prikazan je navedeni filter, čijom se primjenom vidi da skener uistinu uspostavlja vezu te se može utvrditi da je korišten *TCP Connect scan.* U konačnici sve ove informacije pomažu donošenju ključne odluke pri korištenju *Sguil* alata, a to je određivanje je li uzbuna lažni pozitiv, sumnjiva ili maliciozna. Korištenje *Sguila* nije samo kako bi se moglo listati kroz uzbune već te uzbune treba adekvatno klasificirati dodavanje jedne od kategorija incidenata, prikazanih slikom 12.1. Nakon klika na uzbunu potrebno je pritisnuti odgovarajuću F# tipku na tipkovnici (F1 za kategoriju 1, F2 za kategoriju 2, …) nakon čega ona nestaje iz *Sguil* konzole, no i dalje biva spremljena u bazi podataka. Nadalje moguće je dodavati vlastite kategorije jednostavnim unosom u prikazani prozor



Slika12.1 Kategorije incidenata

5. ZAKLJUČAK

Security Onion, besplatna je i otvorenog koda Linux distribucija koja promovira mrežni sigurnosni monitoring (*NSM)* tako što uključuje mnogobrojne sustave, programe i alate koji omogućuju poglede iz raznih kutova na obilje zabilježenog mrežnog prometa.

U jednu ruku sama instalacija ove distribucija ne garantira nam nikakvu sigurnost, ne sprječava nikakve napade i ne mijenja ništa u trenutnoj mrežnoj okolini. Zahtjeva mnogo resursa, što sistemskih, što vremenskih. No s druge strane, niti jedan sigurnosni alat nije savršen niti bez slabosti. Uporni i odlučni provalnici kad-tad će otkriti te slabosti i iskoristiti ih, samo je pitanje vremena.

Ono što nam Security Onion pruža jest vidljivost. Kombinacijom NIDS-a i HIDS-a, od kojih daje više izbora kao što su *Snort* i *Suricata*. *Suricata* IDS nije isproban pa bi trebalo proučiti kako on radi te ga usporediti sa *Snort* IDS-om. Mogućnosti potpunog hvatanja paketa prikazane su alatima *NetworkMiner i Wireshark,* no samo površinski. Oba alata pružaju puno više mogućnosti nego što je prikazano te su vrlo moćni u smislu generiranja podataka o mrežnom prometu. Nadalje *Sguil* alat prikazuje samo uzbune zabilježene već postojećim pravilima te bi trebalo kreirati vlastita pravila koja bi se odnosila na konkretne sumnjive i maliciozne aktivnosti u vlastitoj mreži. Dodatno, u *Sguil* konzoli se još uvijek nalaze nekategorizirane uzbune, no to ne treba obeshrabrivati, NSM je kontinuirani proces, koji se temelji na malim spoznajama i sitnim koracima prema cilju koji je mrežna sigurnsot.

Na prvi pogled može biti zastrašujuće suočiti se sa svim funkcionalnostima, morem informacija i spoznaja o umreženom svijetu te malicioznim aktivnostima koje se događaju svakog trenutka u pozadini, a o kojima do sada nismo ni bili svjesni. No upornošću, strpljivošću i željom za znanjem možemo polako otkrivati kako se suprotstaviti provalnicima i nametnicima te im otežati u njihovim namjerama.

LITERATURA

[1] D. Bourks, Security Onion Documentation, Copyright © Security Onion Solutions, LLC, dostupno na: https://securityonion.readthedocs.io/en/latest/index.html [pristupljeno 20.9.2019]

[2] R. Bejtlich, The Practice of Network Security Monitoring, No Starch Press, Inc, San Francisco,2013

[3]Wireshark Wiki, dostupno na: https://wiki.wireshark.org/Development/LibpcapFileFormat [pristupljeno 20.9.2019.]

[4]help.ubuntu: dostupno na: https://help.ubuntu.com/community/UFW [pristupljeno 20.9.2019.]

[5] https://pastebin.com/YFqNaVi3/ [pristupljeno 20.9.2019.]

[6] D. Bourks, Security Onion github page, dostupno na: https://github.com/Security-Onion-Solutions/security-onion/wiki/tools [pristupljeno 20.9.2019]

[7]Digital Corpora, © 2009-2019 Digital Corpora, dostupno na: https://digitalcorpora.org/corpora/scenarios/nitroba-university-harassment-scenario [pristupljeno 20.9.2019]

[8] LMG Security, LMG Network Forensics Puzzle, dostupno na : http://forensicscontest.com/2010/02/03/puzzle-4-the-curious-mr-x [pristupljeno 20.9.2019.]

SAŽETAK

Upoznati se sa Security Onion Linux distribucijom i njezinim primjenama bio je cilj ovoga završnog rada. Kako bi taj cilj bio ostvaren korištena je *Oracle VM VirtualBox* na kojemu je virtualizirana i osposobljena navedena distribucija. Nadalje, objašnjen je princip mrežnog sigurnosnog monitoringa na kojemu se temelji rad Security Onion Linux distribucije te svakodnevne zadaće i postupci održavanja iste. Proučeno je nekoliko alata koji rade sa potpunim paketima te par scenarija koji demonstriraju neke od primjena. *Tcpdump* i u dijelovima *Tcpreplay*, alatima tekstualnog sučelja, prikazano je manipuliranje *.pcap* datotekama, što su datoteke sa podacima potpunih paketa. Kako ih je moguće spremiti, čitati, ponovno prenositi mrežom. Nadalje *NetworkMiner*, alatom grafičkog sučelja, prikazan je scenarij primanja prijeteće elektroničke pošte te kako je moguće otkriti stvarnu osobu koja se krije iza anonimnosti interneta. NSM konzolom *Sguil* prikazan je proces razlučivanja i kategoriziranja IDS uzbuna vezanih za *port scanove*, jednih od prvih napada koji se događaju i samim time omogućuju provalniku neovlašteni pristup sustavu i osjetljivim informacijama koje se na njemu nalaze. Konkretnih rezultata nije bilo u ovome radu, no stečeno je bolje razumijevanje potencijalnih slabosti i rizika prisutnih u mreži, kao i shvaćanje suparnika koji pokušavaju iskoristiti te slabosti. Također ostvaren je dobar temelj za daljnje poboljšanje sigurnosti i otkrivanje novih načina i rješavanja nepredviđenih problema.

Ključne riječi: Mrežni sigurnosni monitoring, Security Onion Linux distribucija, sustavi za otkrivanje upada, skeniranje portova, Sguil, Snort, NetworkMiner, Wireshark

ABSTRACT

Security Onion Linux distribution and its applications

The goal of this bachelor's thesis was getting familiar with Security Onion Linux distribution and its applications. To achieve this, *Oracle VM VirtualBox* was used, on which the distribution is virtualized and enabled. Furthermore, the concept of Network Security Monitoring (NSM) which is the basis for Security Onion Linux distribution is explained, along with everyday tasks and management chores necessary for optimal performance of the system. A number of tools which either use or generate full packet captures, which are stored in *.pcap* files, together with a couple of scenarios that example the SO applications, are also looked at. *Tcpdump* and *Tcpreplay,* are command-line tools that enable the manipulation of *.pcap* files such as: reading, writing and replaying them over the network. Moreover, *NetowrkMiner,* the GUI tool, shows the harassing e-mail scenario, in which the person hiding behind the anonymity of the Internet, is revealed. Lastly, the NSM console *Sguil* demonstrates the process of categorizing *IDS* alerts generated by port scans, which are first of attacks that the intruder attempts, because they show him the way into the targeted network. This thesis did not have actual results, but it resulted in a better understanding of weaknesses that are present in the network and intruders that want to exploit those weaknesses. Likewise, a good foundation was set for further security upgrades and finding new ways to solve unexpected problems.

Keywords: Network Security Monitoring, Security Onion Linux distribution, port scans, NetworkMiner, Sguil, Snort, Wireshark, IDS

ŽIVOTOPIS

Dominik Bošnjak, rođen je 1997. godine u Požegi gdje pohađa i završava, s odličnim uspjehom Osnovnu Školu Julia Kempfa. Zbog strasti prema Tehnici, koju su mu prenijeli nastavnici u navedenoj osnovnoj školi, upisuje i završava sa istim uspjehom Gimnaziju Požega, prirodoslovno-matematički smjer. Daljnje obrazovanje nastavlja na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Trenutno na navedenom fakultetu završava preddiplomski studij računarstva te planira daljnje obrazovanje nastaviti na istome.